

基礎

2021年9月30日 星期四 下午4:43

remove the object:

```
rm(); rm(list=ls()) # 清除全部
```

Session->clear workspace

Find the detail of a function: `help(xxx)/?xxx`

Change current working directory: tools->global options->general->browse

Change type: `as.numeric('34')`

Repeat: `rep(object, repeat times)`

Sequence: `seq()`

Quit r studio:

Rstudio->quit Rstudio

```
q()
```

Load workspace:

```
Load('FistProject.Rdata')
```

```
Load(file.choose())
```

Session->load workspace

```
Print(); cat()
```

`Print()`: 只能夠輸出單一個 object 的內容，用於輸出 matrix, list 這兩種資料結構的型態非常的方便，不用再額外定義格式

`cat()` :

```
cat("The result of ", x, "plus", y, "is", x+y) The result  
of 3 plus 4 is 7
```

Working Directory

Get current working directory:

```
getwd() # "/Users/jianjiayi/Desktop/數量方法與決策分析"
```

```
setwd('/Users/jianjiayi/Desktop/數量方法與決策分析/youtube')
```

```
setwd('~ /youtube')
```

Change current working directory:

change current working directory.

- tools->global options->general->browse
- Session-> set working directory->choose directory

Vector

Create a vector:

```
c(1, 3, 5, 7, 9) # concatenate
2:7 # sequence
seq(from=1, to=7, by=1) # general sequence
rep(1, times=10) # repeat
rep(seq(from=2, to=5, by=0.25), times=5)
```

```
x = c(1, 3, 5)
x+10 # 11 13 15
```

Extract elements of a vector:

```
x[3] # 5
X[-3] # 1 3
X[1:2] # 1 3
X[c(1,2)] # 1 5
X[-c(1,2)] # 3
X[x<5] # 1 3
```

Matrix

```
mat = matrix(c(1,2,3,4,5,6,7,8,9), nrow=3, byrow=TRUE)
```

```
  [,1] [,2] [,3]
[1,]  1  2  3
[2,]  4  5  6
[3,]  7  8  9
mat[1,2] # 2
mat[c(1,3), 2] # 2 8
mat[2, ] # 4 5 6
mat*10
  [,1] [,2] [,3]
[1,] 10 20 30
[2,] 40 50 60
[3,] 70 80 90
```

Import data from EXCEL(CSV)

```
read.csv(file.choose(), header=T) # header=T 表示第一行是變數名稱
```

```
read.table(file.choose(), header=T, sep=',')  
read.delim(file.choose(), header=T) # Tab-delimited text file  
read.table(file.choose(), header=T, sep='\t')
```

Import dataset->from Excel->sheet: select the sheet you want->NA: 指定特定形式的儲存格為無資料->code preview: next time you want to open this excel with same options, you can just copy this code and paste it in command

Export data

```
write.table(name, file='path', row.names=F, sep=',')  
write.csv(name, file='path', row.names=F)
```

Working with data

```
dim(dataname) # output: row number column number  
head(dataname) # first six rows of the dataset  
tail(dataname)  
Names(dataname)
```

Save workspace

記住所有已經跑過的環境變數

```
Save.image("FirstProject.Rdata")
```

Session->Save workspace as

Dataframe

```
data.frame()  
colnames() <- c()
```

Apply function

2021年10月22日 星期五 上午9:54

```
# Using the 'APPLY' function in R

# read in the "StockExample.csv" data, and attach it
StockData <- read.table(file="/Users/jianjiayi/Desktop/數量方法
與決策分析/youtube/StockExample.csv",
                        sep=";", header=T, row.names=1)

# check the data
StockData

# get the help menu
?apply

# calculate the mean price of each stock
apply(X=StockData, MARGIN=2, FUN=mean)

# calculate the mean price of each stock, removing any NAs
apply(X=StockData, MARGIN=2, FUN=mean, na.rm=TRUE)

# store the mean in an object called AVG
AVG <- apply(X=StockData, MARGIN=2, FUN=mean, na.rm=TRUE)
AVG

# notice that we don't need to include "MARGIN", etc, as long
# as we enter info in the specified order
apply(StockData, 2, mean, na.rm=TRUE)

# do the same, but using the ColMeans command
colMeans(StockData, na.rm=TRUE)

# find the MAXIMUM stock price, for each stock
apply(X=StockData, MARGIN=2, FUN=max, na.rm=TRUE)

# find the 20th and 80th PERCENTILE, for each stock
apply(X=StockData, MARGIN=2, FUN=quantile, probs=c(0.2, .80),
```

```
na.rm=TRUE)
```

```
# create a plot of each column, using a "line"  
apply(X=StockData, MARGIN=2, FUN=plot, type="l")
```

```
# we can also send the plot function more arguments, such as  
# titles, axes labels, and so forth...  
apply(X=StockData, MARGIN=2, FUN=plot, type="l", main="stock",  
      ylab="Price", xlab="Day")
```

```
# now let's calculate the SUM of each row (MARGIN=1)  
apply(X=StockData, MARGIN=1, FUN=sum, na.rm=TRUE)
```

```
# do the same, but with the rowSums command  
rowSums(StockData, na.rm=TRUE)
```

```
# make a nice plot of these...  
plot(apply(X=StockData, MARGIN=1, FUN=sum, na.rm=TRUE),  
     type="l",  
     ,ylab="Total Market Value", xlab="Day", main="Market  
Trend")
```

```
# and add in some nice coloured points...  
points(apply(X=StockData, MARGIN=1, FUN=sum, na.rm=TRUE),  
       pch=16, col="blue")
```

dataframe

2021年10月15日 星期五 上午9:40

將dataframe物件附加到搜尋路徑上，讓我們可以隨時取用該dataframe的所有變數：

```
attach(LungCapData)
```

```
detach() # 養成習慣detach暫時不用的dataframe
```

替代方法：

```
LungCapData$Age
```

```
with(LungCapData, mean(Age))
```

找到dataframe該column下面有哪些內容。若要行使該函數，需在import data的時候加入stringsAsFactors = T，因為只有資料型態為factor才能用該函數。

```
LungCapData <- read.table(file.choose(), stringsAsFactors = T, header=T, sep='\t')
```

```
Levels(Gender) # output: "female" "male"
```

列出所有column的title：

```
name ( )
```

合併dataframe

```
cbind ; rbind
```

```
FemSmoke <- Gender=='female' & Smoke=='Yes'
```

```
MoreData <- cbind(LungCapData, FemSmoke)
```

列出dataframe每一個欄位的基本統計資料

```
summary ( LungCapData )
```

```
any(r_videos_signup[,2] == '簡嘉誼')
```

```
..          瑪莉佳美  0      十一女劉麗君不在此內統計  丁淑筠  1
```


`na.rm`預設值為`false`，表示在判斷是否有指定值時，不計算`na`值

For loop

2021年10月15日 星期五 上午10:54

```
for (i in -3:7) {  
  for (j in 6:9) {  
    print(i^2+sqrt(j))  
  }  
}
```

Install package

2021年10月22日 星期五 上午9:20

```
Install.packages('epiR')
```

如果括號裡面是空白的，R會列出所有的package

括號裡面輸入的package名字要加引號

```
library ( epiR )
```

每次開啟R要使用該package，都要重新load

```
Help(package = epiR)
```

```
Remove.package("epiR")
```

Tools-》install packages

list

2021年10月15日 星期五 上午9:36

list索引：用雙方括號

Eg.

```
ff <- list(dd,c(1,2,3))  
ff[[2]][3] # output: 3
```

tApply function

2021年10月22日 星期五 上午10:11

```
# Using the 'tAPPLY' function in R

# read in the "LungCapData.csv" data, and attach it
LungCapData <-
read.table(file="/Users/jianjiayi/Desktop/數量方法與決策
分析/youtube/LungCapData.txt", sep="\t", header=T)

# check the data
summary(LungCapData)
# and attach it
attach(LungCapData)

# get the help menu
?tapply

# calculate the mean Age for Smoker/NonSmoker
tapply(X=Age, INDEX=Smoke, FUN=mean, na.rm=T)

# you don't need to include "X", "INDEX",... as long as
you
# enter them in that order...
# we also don't need to include "na.rm=T" as no missing
values
tapply(Age, Smoke, mean)

# we can save the output in a new "object"
m <- tapply(Age, Smoke, mean)
m

# also worth discussing is the use of the "SIMPLIFY"
argument
# this is set to TRUE by default...if we set it to
"FALSE"...
tapply(Age, Smoke, mean, simplify=FALSE)
```

```

# note that we could get the same using [ ],
# although using "tapply" is more efficient
mean(Age[Smoke=="no"])
mean(Age[Smoke=="yes"])

# let's look at applying the "summary" function to
groups
tapply(Age, Smoke, summary)

# or, applying the "quantile" function to the groups
tapply(Age, Smoke, quantile, probs=c(0.2, 0.8))

# we can "subset" based on multiple variables/vectors
# calculate the mean Age for Smoker/NonSmoker and
male/female
tapply(X=Age, INDEX=list(Smoke, Gender), FUN=mean,
na.rm=T)

# a less efficient way to get this done...
mean(Age[Smoke=="no" & Gender=="female"])
mean(Age[Smoke=="no" & Gender=="male"])
mean(Age[Smoke=="yes" & Gender=="female"])
mean(Age[Smoke=="yes" & Gender=="male"])

# a reminder of using 2 grouping variables
tapply(Age, list(Smoke, Gender), mean, na.rm=T)

# an a note that the "by" function is the same as
tapply,
# except it presents the results similar to a vector
by(Age, list(Smoke, Gender), mean, na.rm=T)

# and we can subset the elements in the usual way
temp <- by(Age, list(Smoke, Gender), mean, na.rm=T)
temp
temp[4]
# and see the "class" of temp
class(temp)
# we can also convert it to a vector if we prefer
c(temp)
temp2 <- c(temp)
temp2

```

```
## and check it's class  
class(temp2)
```